www.raisecom.com

ACL and Network Security Commands-2

CONTENTS

Chapter 1 ACL and Network Security Commands

1.1 access-list-map

[Function]

Create or delete access-list-map, use this command to enter ACL mapping table.

[Command Format]

access-list-map *<0-399> {permit | deny}*

no access-list-map *<0-399>*

[Parameter]

0-399: serial number for IP Access Control List.

permit: Permit access if conditions are matched.

deny:Deny access if conditions are matched.

[Command Modes]

Global configuration mode; Privileged user.

[Executing Command Instruction]

Use this command to define an IP ACL, the parameter *permit* | *deny* is used to permit or deny the access of packets. This command only set the data filter conditions, and need to be applied to physical port or VLAN to let it be effective.

[Explanation of command execution echo]

access list map 1 is used, can not modify deny or permit.

access list map 1 does not exist

access list map 1 is in use! The operation can't be completed!

[Example]

Raisecom(config)#**access-list-map** *1* **deny**

Raisecom(config-aclmap)#**exit**

Raisecom(config)#**no access-list-map** *1*

[Related commands]

Commands	Description
show access-list-map	Show access-list-map information.

1.2 filter

[Function]

This command is used to add the filter rules. Use **no** form of this command to delete a filter rule. ISCOM 2826/3026 is not in support of the keyword **dougle-tagging**.

[Command format]

[no] filter (ip-access-list | mac-access-list | access-list-map) (all | *{0-399}*) **[double-tagging]**

[no] filter (ip-access-list | mac-access-list | access-list-map) (all | *{0-399})* **(ingress | egress) port-list** *{1-26}* **[double-tagging]**

[no] filter (ip-access-list | mac-access-list | access-list-map) (all | *{0-399}*) **vlan** *<1-4094>*

[no] filter (ip-access-list | mac-access-list | access-list-map) (all | *{0-399}*) **from** *<1-26>* **to** *<1-26>* **[double-tagging]**

[Parameter]

ip-access-list|*mac-access-list*| *access-list-map*: the type of ACL for filtering rule linked list;

all|*{0-399}*: Serial number of ACL, if “all”, it means all defined ACL;

port –list{1-26}: physical port control list;

ingress: filter at the receiving port;

egress: filter at the forwarding port;

from: the filtering receiving port at receiving port and forwarding port;

to: the filtering forwarding port at receiving port and forwarding port;

vlan-list<1-4094>: VLAN number;

double-tagging: filter rule is effective as per double TAG frame format.

[Command mode]

Global configuration mode; Privileged user.

[Executing Command Instruction]

This command is used to add one or more filter rules, the filter rule contains an ordered list of previous defined ACL or VLAN, the priority of these rules is decided by sequence of these filtering rules, the later the filtering rule is added, the higher priority it has. If there is conflicts when the switch tests the packets against the conditions in access list one by one, the higher priority filter rule will be effective (the later added rule). User should properly use all of these rules to limit the incoming packets.

The filter rules will be effective only if filter function is globally enabled.

[Explanation of command execution echo]

Set access list XX unsuccessfully

Delete access list XX unsuccessfully, there is no this filter!

Set successfully

Set unsuccessfully

[Example]

Raisecom(config)#**filter ip-access-list** *0* **ingress portlist** *5*

[Related commands]

Commands	Description
show filter	Show the relevant information for the matching rule filter.
filter enable | disable	Start/cancel the filtering function.

1.3 filter {enable|disable}

[Function]

This command is used to enable filter function globally or disable the filter function.

[Command format]

filter *enable* | *disable*

[Parameter]

enable: Enable filtering function;

disable: Disable filtering function.

[Default]

Disable

[Command Modes]

Global configuration mode; Privileged user.

[Explanation of command execution echo]

Set successfully

Set unsuccessfully

[Example]

Raisecom(config)#**filter** *enable*

[Related commands]

Commands	Description
filter	Add filter rules into rule filter table.
show filter	Show related filter information.

1.4 mac-access-list

[Function]

Set MAC access control list, use “**no**” command to delete.

[Command format]

mac-access-list *<0-399>* **(deny|permit) (ip|arp|rarp|any|***HHHH***)** (*HHHH.HHHH.HHHH* | **any**) (*HHHH.HHHH.HHHH* | **any**)

[Parameter]

0-399: The number of MAC access control list;

permit: permit access if conditions are matched;

deny: deny access if conditions are matched;

protocol: protocol type in the frame head which is denoted by name or numerical value. The protocol type can be **ip, arp, rarp, any**, and the number value is from 0-0xFFFF. If the value is set to *any* or *0*, it stands for all the protocols;

HHHH.HHHH.HHHH | ***any***: source MAC address, adopt dotted hexadecimal numeral, two characters for a group, any stands for any source MAC address;

HHHH.HHHH.HHHH | ***any***: destination MAC address, adopt dotted hexadecimal numeral, two characters for a group, any stands for any destination MAC address.

[Command Modes]

Global configuration mode; privileged user.

[Executing Command Instruction]

Use this command to define a MAC ACL, parameter *permit* | *deny* is used to set the switch whether to permit or deny the access of the packet. This command is only used to set the filter rule, generally speaking, and it should be applied to physical port or VLAN to be effective.

[Explanation of command execution echo]

Set successfully

Set unsuccessfully!

[Example]

Raisecom (config)#**mac-access-list** *10* **deny any** *1234.1234.1234 1111.2222.3344*

[Related commands]

Commands	Description
no mac-access-list (*{0-399}***|*all*)**	Delete the speicified MAC access control list.
show mac-access-list [{*0-399*}]	Show information of specified MAC access control list.

1.5 match(CMAP)

[Function]

This command is used to define Traffic Classification.

[Command format]

match { **ip-access-list** *acl-index* | **mac-access-list** *acl-index* | **access-list-map** *acl-index* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **class** *calss-name* | **vlan** *vlanlist* [**double-tagging inner**] }

no match { **ip-access-list** *acl-index* | **mac-access-list** *acl-index* | **access-list-map** *acl-index* | **ip dscp | ip precedence | class** *calss-name* | **vlan** *vlanlist* }

[Parameter]

ip-access-list *acl-index*: specify the number of IP ACL.

mac-access-list *acl-index*: specify the number of MAC ACL.

access-list-map *acl-index*: specify user defined number of ACL.

ip dscp *dscp-list*: specify DSCP value for incoming packets, the range is from 0 to 63.

ip precedence *ip-precedence-list*: specify IP priority range from 0 to 7.

calss *calss-map*: specify a class map, this classmap can only be the type of match-all.

vlan *vlanlist*: specify vlan id, range from 1 to 4094.

double-tagging inner: match with inner VLAN TAG.

[Command Modes]

CMAP configuration mode; Privileged user.

[Executing Command Instruction]

match is used to define the traffic classification under the class-map configuration mode. Be attention that there maybe conflict among different matching types when classify incoming packets. When use previous defined ACL entries for classification, ACL type should be **permit**.

[Explanation of command execution echo]

Set the match statement for the class map successfully.

Set the match statement for the class map unsuccessfully.

The input parameter error.

The input name is too long.

[Example]

Raisecom(config)# **ip-access-list** *1* **permit any any dscp** *10*

Raisecom(config)# **class-map** *class 1 match-all*

Raisecom(config-cmap)# **match ip-access-list** *1*

Raisecom(config-cmap)# **no match ip-access-list** *1*

[Related commands]

Commands	Description
show class-map [*class-map-name***]**	Show class-map information.

1.6 match(ACLMAP layer 2)

[Function]

Define the ACL layer-2 head data matching.

[Command format]

match mac *{destination*|*source}* *HHHH.HHHH.HHHH*

match cos *<0-7>*

match ethertype *HHHH [HHHH]*

match *{arp* | *eapol* | *flowcontrol* | *ip* | *ipv6* | *loopback* | *mpls* | *mpls-mcast* | *pppoe* | *pppoedisc* | *x25* | *x75}*

no match mac *{destination*|*source}*

no match cos

no match ethertype

[Parameter]

mac: match layer 2 MAC address.

destination: match layer 2 destination MAC address.

source: match layer 2 source MAC address.

HHHH.HHHH.HHHH: MAC address.

cos: match cos value

ethertype: match the protocol type of layer 2

arp: match ARP

eapol: match eapol

flowcontrol: match flowcontrol

ip: match ip

ipv6: match ipv6

loopback: match loopback

mpls: match mpls unitcast protocol.

mpls-mcast: match mpls multicast protocol.

pppoe: match pppoe

pppoedisc: match pppoe discovery protocol

x25: match x25 protocol.

x75: match x75 protocol.

[Command Modes]

ACLMAP configuration mode; privileged user.

[Executing Command Instruction]

Match is used to define the match conditions of user define access-list under access-list-map. With this command our users can define the layer-2 ACL entries flexibly, and all the first 64 bytes can be set as the match conditions.

[Explanation of command execution echo]

Conflict with previous matches.

[Example]

Raisecom(config)# **access-list-map** *101 deny*

Raisecom(config-aclmap)# **match mac *destination*** *000e.5e11.2344*

Raisecom(config-aclmap)# **match cos** *3*

Raisecom(config-aclmap)# **match ethertype** *0800 ff00*

Raisecom(config-aclmap)# **match *ipv6***

Raisecom(config-aclmap)# **no match *cos***

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.7 match arp

[Function]

Use this command to define arp data matching of map table for ACL.

[Command format]

match arp opcode *{request* | *reply}*

match arp sender-mac *HHHH.HHHH.HHHH*

match arp target-mac *HHHH.HHHH.HHHH*

match arp sender-ip *A.B.C.D [A.B.C.D]*

match arp target-ip *A.B.C.D [A.B.C.D]*

no match arp opcode

no match arp sender-mac

no match arp target-mac *HHHH.HHHH.HHHH*

no match arp sender-ip

no match arp target-ip

[Parameter]

opcode: match ARP packet type.

request: match arp request packet.

reply: match arp reply packet.

sender-mac: match mac address of ARP sender.

target-mac: match ARP target hardware address.

HHHH.HHHH.HHHH: MAC address.

sender-ip: match IP address of ARP sender.

target-ip: match ARP target IP address.

ethertype: match layer 2 protocol type

A.B.C.D [A.B.C.D]: IP address (mask)

[Command mode]

ACLMAP configuration mode; Privileged user.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define arp protocol match conditions. **Note**: there may be conflict during matching different types.

[Explanation of command execution echo]

Conflict with previous matches.

[Example]

Raisecom(config)# **access-list-map** *101 deny*

Raisecom(config-aclmap)# **match arp opcode** ***request***

Raisecom(config-aclmap)# **match sender-mac** *000e.5e23.4553*

Raisecom(config-aclmap)# **match sender-ip** *10.0.0.0 255.0.0.0*

Raisecom(config-aclmap)# **no match arp opcode**

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.8 match ip

[Function]

Use this command to define ip protocol data matching of map table for ACL.

[Command format]

match ip ***{destination-address*** **|** ***source-address}*** *A.B.C.D [A.B.C.D]*

match ip precedence {*<0-7>* **|** ***routine*|** ***priority*|** ***immediate*|** ***flash*|** ***flash-override*** **|** ***critical*** **|** ***internet*** **|** ***network*}**

match ip tos {*<0-15>* **|** ***normal*** **|** ***min-monetary-cost*** **|** ***min-delay*** **|** ***max-reliability*** **|** ***max-throughput*}**

match ip dscp {*<0-63>* **|** ***af11*** **|** ***af12*** **|** ***af13*** **|** ***af21*** **|** ***af22*** **|** ***af23*** **|** ***af31*** **|** ***af32*** **|** ***af33*** **|** ***af41*|** ***af42***

|*af43* | *cs1* | *cs2* | *cs3* | *cs4* | *cs5* | *cs6* | *cs7* | *ef* | *default*}

match ip no-fragments

match ip {*ahp* | *esp* | *gre* | *icmp* | *igmp* | *igrp* |*ipinip* | *ospf* | *pcp* | *pim* | *tcp* | *udp*}

match ip protocol *<0-255>*

no match ip *{destination-address* | *source-address}*

no match ip precedence

no match ip tos

no match ip dscp

no match ip no-fragments

no match ip protocol

[Parameter]

destination-address: match IP target address.

source-address: match IP source address.

precedence: match IP priority

<0-7>: IP priority value

routine: IP priority value is 0

priority: IP priority value is 1

immediate: IP priority value is 2

flash: IP priority value is 3

flash-override: IP priority value is 4

critical: IP priority value is 5

internet: IP priority value is 6

network: IP priority value is 7

tos: match IP TOS value

<0-15>: TOS value

normal: normal TOS value(0)

min-monetary-cost: minimum monetary cost TOS value (1)

min-delay: minimum delay TOS value（8）

max-reliability: maximum reliable TOS value（2）

max-throughput: maximum throughput rateTOS value（4）

dscp: match IP dscp value.

<0-63>: ip dscp value.

af11: AF11 dscp value（001010）

af12: AF12 dscp value（001100）

af13: AF13 dscp value （001110）

af21: AF21 dscp value（010010）

af22: AF22 dscp value（010100）

af23: AF23 dscp value（010110）

af31: AF31 dscp value（011010）

af32: AF32 dscp value（011100）

af33: AF33 dscp value（011110）

af41: AF41 dscp value （100010）

af42: AF42 dscp value（100100）

af43: AF43 dscp value（100110）

cs1: CS1(priority 1) dscp value（001000）

cs2: CS2(priority 2) dscp value（010000）

cs3: CS3(priority 3) dscp value（011000）

cs4: CS4(priority 4) dscp value（100000）

cs5: CS5(priority 5) dscp value（101000）

cs6: CS6(priority 6) dscp value（110000）

cs7: CS7(priority 7) dscp value（111000）

default: default dscp value（000000）

ef: EF dscp value（101110）

no-fragments: match no-fragments packet

protocol: match IP protocol type.

<0-255>: P protocol type value.

ahp: Authentication Header protocol

esp: encapsulation security protocol

gre: general router encapsulation protocol

icmp: Internet Control Message Protocol

igmp: Internet Group message protocol

igrp: Interior gateway protocol

ipinip: IP-in-IP tunnel

ospf: Open Shortest-Path First

pcp: IP Payload Compression protocol

pim: Protocol Independent Multicast protocol

tcp: Transmission Control Protocol

udp: User Datagram Protocol

[Command format]

ACLMAP configuration mode; privileged user.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define IP protocol match conditions. Note: there may be conflict during matching different types. ToS or IP precedence and dscp confliction.

[Explanation of command execution echo]

Conflict with previous matches.

[Example]

```
Raisecom(config)# access-list-map 101 deny
Raisecom(config-aclmap)# match ip destination-address 10.1.23.4.5
Raisecom(config-aclmap)# match ip precedence priority
Raisecom(config-aclmap)# match ip tos normal
Raisecom(config-aclmap)# match ip dscp 34
Raisecom(config-aclmap)# match ip no-fragments
Raisecom(config-aclmap)# match ip no-fragments
Raisecom(config-aclmap)# match ip ospf
Raisecom(config-aclmap)# no match ip protocol
```

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.9 match ip icmp

[Function]

Define icmp protocol match conditions.

[Command format]

match ip icmp *<0-255> [<0-255>]*

[Parameter]

<0-255> [<0-255>]: ICMP message type.

[Command format]

ACLMAP configuration mode; privileged user.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define IP ICMP protocol match conditions. Pay attention to the conflict among different types.

[Explanation of command execution echo]

Conflict with previous matchs.

[Example]

Raisecom(config)# **access-list-map** *101 deny*

Raisecom(config-aclmap)# **match ip icmp** *2 2*

Raisecom(config-aclmap)# **no match ip protocol**

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.10 match ip igmp

[Function]

Use this command to define the IGMP protocol match condition.

[Command format]

match ip igmp {*<0-255>* | *dvmrp* | *query* | *leave-v2* | *report-v1* | *report-v2* |*report-v3* | *pim-v1*}

[Parameter]

<0-255>: IGMP message type

dvmrp: Distance Vector Multicast Routing Protocol

leave-v2: IGMPv2 leave group

pim-v1: protocol individual message version 1

query: IGMP member query

report-v1: IGMPv1 member report

report-v2: IGMPv2 member report

report-v3: IGMPv3 member report

[Command Modes]

ACLMAP configuration mode;Privileged user.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define IP IGMP protocol match conditions.

[Explanation of command execution echo]

conflict with previous matchs.

[Example]

```
Raisecom(config)# access-list-map 101 deny
Raisecom(config-aclmap)# match ip igmp query
Raisecom(config-aclmap)# no match ip protocol
```

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.11 match ip tcp

[Function]

Define the tcp protocol match conditions for ACL.

[Command Format]

match ip tcp { destination-port | source-port} {*<0-65535>* **| bgp | domain | echo | exec | finger | ftp | ftp-data | gopher | hostname | ident | irc | klogin | kshell | login | lpd | nntp | pim-auto-rp | pop2 | pop3 | smtp | sunrpc | syslog | tacacs | talk | telnet | time | uucp | whois | www}**

match ip tcp {ack | fin | psh | rst | syn | urg }

no match ip tcp { destination-port | source-port}

no match ip tcp {ack | fin | psh | rst | syn | urg }

[Parameter]

destination-port: match ip tcp Destination Port

source-port: match ip tcp source port

<0-65535>: tcp port number

bgp: Border Gateway Protocol（179）

domain: Domain Name Service（53）

echo: Echo protocol（7）

exec: Exec (rsh, 512)

finger: Finger (79)

ftp: file transmission protocol（21）

ftp-data: FTP data connection（20）

gopher: Gopher (70)

hostname: NIC hostname server (101)

ident: identification protocol (113)

irc: IRC protocol (194)

klogin: Kerberos login (543)

kshell: Kerberos shell (544)

login: Login (rlogin, 513)

lpd: printer service protocol(515)

nntp: Network News Transfer Protocol

pim-auto-rp: PIM Auto-RP (496)

pop2: Post Office Protocol Version 2(109)

pop3: Post Office Protocol Version 3 (110)

smtp: Simple Mail Transfer Protocol (25)

sunrpc: Remote Procedure Call protocol (111)

syslog: system log (514)

tacacs: TAC Acquisition and Control System (49)

talk: Talk (517)

telnet: Telnet (23)

time: Time (37)

uucp: Unix-to-Unix copy program (540)

whois: Nicname(43)

www: World Wide Web (HTTP, 80)

ack: match ACK

fin: match FIN

psh: match PSH

rst: match RST

syn: match SYN

urg: match URG

[Command format]

ACLMAP configuration mode; privileged user.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define TCP protocol match conditions.

[Explanation of command execution echo]

conflict with previous matchs.

[Example]

```
Raisecom(config)# access-list-map 101 deny
Raisecom(config-aclmap)# match ip tcp destination-port smtp
Raisecom(config-aclmap)# match ip tcp source-port 6201
Raisecom(config-aclmap)# match ip tcp ack
Raisecom(config-aclmap)# match ip tcp fin
Raisecom(config-aclmap)# no match ip tcp destination-port
Raisecom(config-aclmap)# no match ip tcp fin
```

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.12 match ip udp

[Function]

Use this command to define udp protocol match conditions.

[Command format]

match ip udp { destination-port | source-port } {*<0-65535>* **| biff | bootpc | bootps | domain | echo | mobile-ip | netbios-dgm | netbios-ns | netbios-ss | ntp | pim-auto-rp | rip | snmp | snmptrap | sunrpc | syslog | tacacs | talk | tftp | time | who }**

no match ip udp { destination-port | source-port}

[Parameter]

destination-port: match ip udp destination port

source-port: match ip udp source port

<0-65535>: udp port number

biff: Biff (mail notification, comsat, 512)

bootpc: boot protocol(BOOTP)client end（68）

bootps: boot protocol(BOOTP)server end（67）

domain: domain service protocol（53）

echo: echo protocol（7）

mobile-ip: mobile IP registration (434)

netbios-dgm: NetBios data message service（138）

netbios-ns: NetBios name service（137）

netbios-ss: NetBios session service（139）

ntp: Network Time Protocol (123)

pim-auto-rp: PIM Auto-RP (496)

rip: router information protocol(520)

snmp: Simple Network Management Protocol (161)

snmptrap: SNMP Traps (162)

sunrpc: Sun remote process control(111)

syslog: system log(514)

tacacs: TAC access control system (49)

talk: Talk (517)

tftp: Trivial File Transfer Protocol (69)

time: Time (37)

who: Who service (rwho, 513)

[Command Modes]

ACLMAP configuration mode; privileged use exec.

[Executing Command Instruction]

Under access-list-map configuration mode, **match** command is used to define UDP protocol match conditions.

[Explanation of command execution echo]

Conflict with previous matchs.

[Example]

Raisecom(config)# **access-list-map** *101 deny*

Raisecom(config-aclmap)# **match ip** ***udp destination-port tacacs***

Raisecom(config-aclmap)# **match ip** ***udp source-port 7306***

Raisecom(config-aclmap)# **no match ip** ***udp destination-port***

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.
show access-list-map [*acl-index*]	Show access-list-map information.

1.13 match user-define

[Function]

Define the user defined match conditions.

[Command format]

match user-define *RULE-STRING RULE-MASK <0-64>*

no match user-define

[Parameter]

MATCH-STRING: match data, hex string;

RULE-MASK: mask of match data, used to filter match data from incoming packets.

<0-64>: Location of the matching data that offsets from header of L2 frame. For untag packets, please remember that switch will add 4 bytes (IEEE802.1Q tag) and set the offset of matching data carefully.

[Command Modes]

ACLMAP configuration mode;Privileged user.

[Executing Command Instruction]

Access-list-map configuration mode, **match user-define** command is for users to define matching conditions by themselves. It is very flexible for user to define the ACL entries when the incoming packets are not in regular frame structure.

[Explanation of command execution echo]

Length of match data and mask is not equal!

The match data overrun the frame!

The match data is INVALID!

The mask data is INVALID!

[Example]

Raisecom(config)# **access-list-map** *101 deny*

Raisecom(config-aclmap)# **match user-define** ***a0 ff 24***

Raisecom(config-aclmap)# **no match user-define**

[Related commands]

Commands	Description
show access-list-map [*acl-index*]	Show access-list-map information.

1.14 show access-list

[Function]

This command is used to show the ACL information.

[Command format]

show (ip-access-list|mac-access-list) *[{0-399}]*

[Parameter]

ip-access-list|mac-access-list:The ACL type used by filtering rule.

{0-399}:Serial number of ACL, if the parameter is ignored, then that is the all the defined ACL.

[Command Modes]

Global configuration mode; privileged user.

[Executing Command Instruction]

This command is used to show the ACL information.

[Explanation of command execution echo]

Show the type of ACL, time for which is cited by the filtering rule, actual number of matching rule and other parameters.

[Example]

Show ip-access-list

Show mac-access-list *0-5*

[Related commands]

Commands	Description
access-list	Relevant ACL
no access-list	Delete relevant ACL table.

1.15 show access-list-map

[Function]

This command is used to show ACL map table configured content for relevant type.

[Command format]

Show access-list-map *[0-399]*

[Parameter]

access-list-map:ACL map table

{0-399}:Serial number of ACL, if the parameter is ignored, then that is the all the defined ACL.

[Command Modes]

Global configuration mode; privileged user.

[Executing Command Instruction]

This command is used to show the configured content of ACL.

[Explanation of command execution echo]

Show the actual matching rule of ACL map.

[Example]

show access-list-map *10*

[Related commands]

Commands	Description
access-list-map	Define related ACL map table.
no access-list-map	Delete related ACL map table.

1.16 show filter

[Function]

This command is used to show the related information of filter.

[Command format]

show filter

[Command Modes]

Privileged EXEC

[Executing Command Instruction]

This command is used to show the related information of the filter. The content is shown based on the order of arrival, the earlier the ACL is added, the more frontal it is.

[Explanation of command execution echo]

```
Rule filter: Disable

Filter list(Larger order number, Higher priority):

Order ACL-Index    IPort    EPort VLAN Hardware

-------------------------------------------------------------------

1        MAP    0        1        -         -      No

2        IP       0       -        3         -      No
```

[Example]

show filter

[Related commands]

Commands	Description
filter	Put the filter rule into the rule filter table.
filter *enable* | *disable*	Start or cancel filter function.

北京瑞斯康达科技发展有限公司
RAISECOM TECHNOLOGY CO.,LTD.

Address: 2nd Floor, South Building of Rainbow Plaza, No.11 Shangdi Information Road, Haidian District, Beijing Postcode: 100085 Tel: +86-10-82883305 Fax: +86-10-82883056
Email: export@raisecom.com http://www.raisecom.com